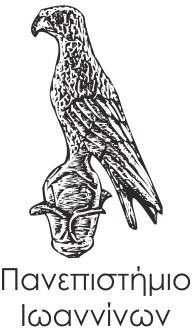
**ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ**



**ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ**

**ΑΛΓΟΡΙΘΜΟΙ ΚΑΙ ΠΡΟΧΩΡΗΜΕΝΕΣ ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ**

**Διερεύνηση προσεγγίσεων επίλυσης του προβλήματος 0-1 σακιδίου**

ΤΕΧΝΙΚΗ ΕΚΘΕΣΗ

**ΛΟΥΠΑΣ ΒΑΣΙΛΕΙΟΣ**

**ΑΜ: 59**

Ημερομηνία Υποβολής: 7/1/2020

**1. ΤΟ ΠΡΟΒΛΗΜΑ**

Διερεύνηση προσεγγίσεων επίλυσης του προβλήματος 01σακιδίου (0/1 Knapsack)

Το πρόβλημα του 0/1 σακιδίου (01 knapsack) αφορά ένα σύνολο από αντικείμενα για τα οποία γνωρίζουμε το βάρος και την αξία κάθε αντικειμένου.

Ζητείται η επιλογή ενός υποσυνόλου των αντικειμένων έτσι ώστε το συνολικό βάρος από τα επιλεχθέντα αντικείμενα να μην ξεπερνά μια συγκεκριμένη τιμή βάρους και ταυτόχρονα

να επιτυγχάνεται η μεγαλύτερη δυνατή αξία.

Το πρόθεμα 0/1στο όνομα του προβλήματος υποδηλώνει ότι κάθε αντικείμενο μπορεί είτε να επιλεχθεί είτε να μην επιλεχθεί στο σύνολό του και όχι τμηματικά.

**2. ΠΡΟΣΕΓΓΙΣΕΙΣ ΕΠΙΛΥΣΗΣ - ΜΕΘΟΔΟΛΟΓΙΑ**

Για να διερευνήσουμε τις προσεγγίσεις επίλυσης του προβλήματος που μας έχει ανατεθεί πρέπει πρώτα να φτιαχτούν τα αρχεία (στιγμιότυπα) των προβλημάτων με τον generator που μας δόθηκε. Κατεβάζουμε το generator.c από τη διεύθυνση <http://hjemmesider.diku.dk/~pisinger/generator.c> και το κάνουμε compile (g++ -o generator generator.c) σε ένα terminal στο linux.

Φτιάχνουμε ένα linux bash file **(makefiles)** το οποίο εκτελούμε και παράγουμε τα 320 αρχεία με βάση την ονοματολογία που μας ζητήθηκε.Τα αρχεία αυτά έχουν την μορφή που παρουσιάζεται παρακάτω:

10

1 4 4

2 46 46

3 48 48

4 19 19

5 4 4

6 20 20

7 13 13

8 50 50

9 28 28

10 6 6

79

Το πρώτο νούμερο αναφέρεται στο πλήθος των αντικειμένων που μελετούνται, η πρώτη στήλη είναι ο Αύξων Αριθμός του αντικειμένου, η δεύτερη στήλη είναι η αξία του αντικειμένου και η τρίτη στήλη είναι το βάρος τους. Το νούμερο της τελευταίας γραμμής φανερώνει τη συνολική χωρητικότητα του σακιδίου.

Έπειτα φτιάχτηκαν δυο προγράμματα σε c++που οποίο ανοίγουν τα αρχεία αυτά ένα ένα και καλεί functions οι οποίες υλοποιούν διαφορετικές μέθοδοι/αλγόριθμους επίλυσης, τους:

* Dynamic Programming
* Greedy Approach
* Brute Force
* Branch and Bound
* OR-TOOLS

και επιστρέφει σαν αποτέλεσμα την αξία (value) την οποία υπολόγισε η κάθε μέθοδος, καθώς και τον χρόνο που χρειάστηκε (σε nanoseconds οι 4 πρώτες και millisecond η OR-TOOLS) ώστε να παράγει το αποτέλεσμα.

Ο κώδικας **Knapsack-Code.cc** και ο **ORTOOLS-Knap.cc** όταν εκτελεστούν παράγουν από ένα .csv αρχείο με το όνομα **results.csv** και **or-results.csv.**

Έπειτα τα αρχεία .csv ενώθηκαν και εισάχθηκε σε ένα book του Excel και προέκυψαν ραβδογράμματα για τις αξίες των λύσεων που μας έδωσαν οι 5 διαφορετικές προσεγγίσεις.

Τα ραβδογράμματα αυτά προέκυψαν ομαδοποιώντας τα αποτελέσματα ανά αρχείο στιγμιοτύπων (Πρώτα τα στιγμιότυπα με ν=10, έπειτα με ν=50, έπειτα με ν=100 και τέλος με ν=500).

Το κάθε ραβδόγραμμα περιέχει 80 στήλες οι οποίες χωρίζονται σε 4 τιμές (1 για κάθε μέθοδο).

